**Generări de tip backtracking**

Ex: Să generăm toate cuvintele formate din 4 vocale distincte, în care vocalele apar în ordine alfabetică (Vocalele fiind A E I O U)

Soluțiile:

AEIO

AEIU

AEOU

AIOU

EIOU

**1) Generare de permutări**

Permutări de n: toate posibilitățile de a pune într-un vector cu n elemente numerele de la 1 la n fără a le repeta (vectorul, d.p.d.v. matematic, reprezintă o "mulțime ordonată". Prin "mulțime ordonată" se înțelege o mulțime de elemente în care contează ordinea. Mulțimile clasice (cele din clasele 1-8) sunt neordonate, adică nu contează ordinea scrierii. Ele se scriu între acolade, pe când mulțimile ordonate se scriu între paranteze rotunde.)

Ex:

Iată permutările de 4:

(1, 2, 3, 4) (2, 1, 3, 4) (3, 1, 2, 4) (4, 1, 2, 3)

(1, 2, 4, 3) (2, 1, 4, 3) (3, 1, 4, 2) (4, 1, 3, 2)

(1, 3, 2, 4) (2, 3, 1, 4) (3, 2, 1, 4) (4, 2, 1, 3)

(1, 3, 4, 2) (2, 3, 4, 1) (3, 2, 4, 1) (4, 2, 3, 1)

(1, 4, 2, 3) (2, 4, 1, 3) (3, 4, 1, 2) (4, 3, 1, 2)

(1, 4, 3, 2) (2, 4, 3, 1) (3, 4, 2, 1) (4, 3, 2, 1)

**Tipic pentru permutări**

- elementele NU se repetă

- o permutare cu n elemente folosește TOATE cele n elemente ale unei mulțimi date.

Numărul lor P**n** = n!

**2) Generare de aranjamente ale unei mulțimi cu n elemente, luate câte m**

Aranjamente de n luate câte m : toate posibilitățile de a pune într-un vector cu DOAR m elemente numerele de la 1 la n (unde m≤n) fără a repeta valori în același vector. Din nou avem de-a face cu mulțimi ordonate.

Ex: Iată aranjamentele de 5 luate câte 3:

(1, 2, 3) (2, 1, 3) ...... (5, 1, 2)

(1, 2, 4) (2, 1, 4) (5, 1, 3)

(1, 2, 5) (2, 1, 5) (5, 1, 4)

(1, 3, 2) (2, 3, 1) (5, 2, 1)

(1, 3, 4) (2, 3, 4) (5, 2, 3)

(1, 3, 5) (2, 3, 5) (5, 2, 4)

(1, 4, 2) (2, 4, 1) (5, 3, 1)

(1, 4, 3) (2, 4, 3) (5, 3, 2)

(1, 4, 5) (2, 4, 5) (5, 3, 4)

(1, 5, 2) (2, 5, 1) (5, 4, 1)

(1, 5, 3) (2, 5, 3) (5, 4, 2)

(1, 5, 4) (2, 5, 4) (5, 4, 3)

Numărul lor:

**Tipic pentru aranjamente**

- elementele NU se repetă

- o soluție particulară dintre aranjamente folosește DOAR m elemente dintre cele n date

- ordinea scrierii e importantă: Dacă schimbăm ordinea a două elemente, se obține o cu totul altă soluție.

**3) Generare de combinări ale unei mulțimi cu n elemente, luate câte m**

Combinări de n luate câte m : toate posibilitățile de a pune într-o mulțime neordonată cu DOAR m elemente numerele de la 1 la n (unde m≤n) fără a repeta valori. De data asta avem de-a face cu mulțimi neordonate (obișnuite). De regulă, pentru a NU repeta valori, se convine ca elementele unei anumite soluții particulare să fie scrise în ordine crescătoare (sau mai rar, descrescătoare)

Ex: Iată combinările de 6 luate câte 4:

{1, 2, 3, 4} {2, 3, 4, 5} {3, 4, 5, 6}

{1, 2, 3, 5} {2, 3, 4, 6}

{1, 2, 3, 6} {2, 3, 5, 6}

{1, 2, 4, 5} {2, 4, 5, 6}

{1, 2, 4, 6}

{1, 2, 5, 6}

{1, 3, 4, 5}

{1, 3, 4, 6}

{1, 3, 5, 6}

{1, 4, 5, 6}

Numărul lor:

**Tipic pentru combinări**

- elementele NU se repetă

- o soluție particulară dintre combinări folosește DOAR m elemente dintre cele n date

- ordinea scrierii NU e importantă: dacă schimbăm ordinea a două elemente, se obține de fapt aceeași soluție. Din acest motiv, elementele unor combinări apar de regulă în ordine fie crescătoare, fie descrescătoare.

**4) Generare elemente ale unui produs cartezian**

Produs cartezian dintre niște mulțimi M1, M2, M3, ... Mn  este totalitatea mulțimilor ordonate de elemente x1, x2, ..., xn în care x1 ∈ M1, x2 ∈ M2, xn ∈ Mn fără absolut nicio restricție.

Mai mult, mulțimile M1, M2, ... se pot repeta, pot fi chiar una și aceeași mulțime.

Ex:

Dacă

M1={t,h}

M2={Δ, Ω, Ψ}

M3={•, ⊥}

Elementele produsului cartezian M1xM2xM3 sunt:

(t, Δ, •) (t, Δ, ⊥) (t, Ω, •) (t, Ω, ⊥) (t, Ψ, •) (t, Ψ, ⊥)

(h, Δ, •) (h, Δ, ⊥) (h, Ω, •) (h, Ω, ⊥) (h, Ψ, •) (h, Ψ, ⊥)

Numărul total de elemente din produsul cartezian este date de produsul numărului de elemente din toate mulțimile. La noi 2\*3\*2=12.

**Tipic pentru produs cartezian**

- elementele se pot repeta (nu au nicio restricție) evident, în cazul în care mulțimile pe care se face produsul cartezian sunt aceleași.

Exemplu: Generarea tuturor codurilor PIN de 4 cifre reprezintă un produs cartezian al mulțimii M={0,1,2,3,4,5,6,7,8,9} cu ea însăși de 4 ori.

**Exemplu de exercițiu de simulare a unei generări:**

Să se genereze în ordine lexicografică toate numerele cu cifre **distincte**, astfel încât suma cifrelor să fie egală cu 3.

102

12

120

201

21

210

3

30

**Exercițiu**

Să generăm submulțimile nevide ale mulțimii {1,2,3,4} în ordine lexicografică

1

1 2

1 2 3

1 2 3 4

1 2 4

1 3

1 3 4

1 4

2

2 3

2 3 4

2 4

3

3 4

4

(Ne reamintim că o mulțime cu n elemente are un număr total de 2n submulțimi, incluzând‑o și pe cea vidă)